공학석사 학위논문

음향 측정 데이터를 이용한 산업용 검사를 위한 컨벌루션 신경망

Convolutional Neural Networks for Industrial Inspection Using Sound Measurement Data

2017년 8월

서울대학교 대학원

기계항공공학부

송 지 훈

ABSTRACT

Convolutional Neural Networks for Industrial Inspection Using Sound Measurement Data

by

Jihoon Song

School of Mechanical and Aerospace Engineering
Seoul National University

This thesis proposes an inspection method using a convolutional neural network (CNN) to automate industrial inspection using sound measurement data. We first consider the industrial inspection problem as a classification problem in machine learning to automate inspection. Given the sound measurement data of normal and defective samples for rotating machines, which can be inspected with sound measurement data, we train a classifier that use the CNN, which is a kind of deep learning. In general, it is difficult to obtain large amounts of data for learning in industrial inspection problems. To overcome the lack of training data, we use transfer learning.

In addition, Greedy layer-wise supervised training method is proposed to improve the performance in transfer learning. As an example of industrial inspection using sound measurement data, we conduct the inspection of the electric motor used in the drones by the inspection method presented above. Given the sound measurement data of the electric motors, we perform several experiments to show the performance of our algorithm. Our inspection algorithm using the CNN shows better detection of defective motors than the inspection using conventional classification method in machine learning. Especially, the algorithm using the CNN is a kind of end-to-end learning, and it shows excellent performance without manually extracting the features adequate to the given data. Therefore, it is applicable to various inspection fields using sound measurement data without deep understanding of the given data.

Keywords: convolutional neural network, transfer learning, Greedy layer-wise supervised training, electric motor inspection, end-to-end learning

Student Number: 2015-20734

Contents

List of Tables

List of Figures

1

Introduction

Fault inspection is one of the essential production process in industrial areas. It is also very important to automate the fault detection process. If the inspection process is not automated, factory workers must manually inspect a lot of products. These simple repetitive tasks of human workers take a long time, ultimately lead to their fatigue, and cause their mistakes. If defective products are included in the final products due to mistakes, the company's reputation may become unfavorable and result in financial loss. Thus the automated industrial inspection is indispensable part of manufacturing processes.

Especially, for rotating machines such as engines or motors, it may be more useful to detect faults of products with sound measurement data than with visual data. Such rotating machines are composed of a complex combination of small parts inside, so it is not easy to determine whether there are defects in visual inspection. These rotating machines usually make sounds while spinning. If there are defective parts in the products, it will sound a little different. Because of this characteristic, it is easier to identify the defect as sound measurement data.

1.1 Previous Research

Fault Inspection using Sound Measurement Data

Many researchers have used sound measurement data to diagnose defects in rotating machines. Among these studies, there is an attempt to inspect faults in a rotating machine using a symmetrised dot pattern (SDP) method to visualize a sound signal[2]. Some researchers particularly conduct studies on fault inspection of motors. One of them diagnoses fault of induction motor by applying smoothed pseudo wigner-ville distribution (SPWVD) to stator current and vibration as well as sound signal[3]. Another researches on motor inspection are to apply Hilbert transform[4] or wavelet transform[5] to the sound signal to diagnose the fault of motors. In addition, the Kohonen self-organizing map is applied to the sound to diagnose induction motor faults[6]. Moreover, there is an attempt to classify internal combustion engines using an artificial neural network by applying discrete wavelet transform (DWT) or continuous wavelet transform (CWT) to the sound emission signals[7, 8].

All of these studies attempt to extract certain features from sound measurement data. From these features, it is judged whether or not rotating machines is defective. Extracting features from the data and then detecting defects with these features is a way to approach common inspection problems. This process is similar to conventional approaches to the classification problem of machine learning. Therefore, inspection problem using sound measurement data can be regarded as a sort of classification problem.

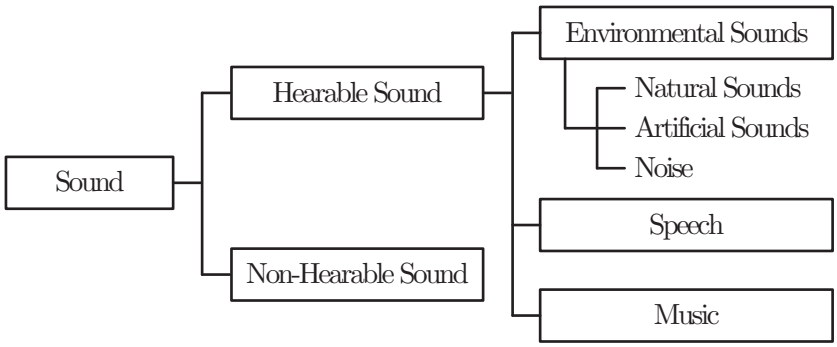


Figure 1.1: A taxonomy of sound (adapted from [1])

Conventional Approch to Sound Classification

In general, sounds can be classified according to their characteristics as shown in Figure 1.1[1]. Depending on the type of sound, the approach to the classification problem is different. The sound from rotating machines can be seen as an environmental sound in Figure 1.1, which has several distinct characteristics from speech and music.

First, speech and music are represented by sound units, phonemes and notes, respectively. On the other hand, the environmental sound has no specific sound unit and its expression is theoretically infinite. This characteristic of the environmental sound makes it difficult to extract features from sound measurement data. In [9], we can see a summary of many features of sound. Frequently used features include mel-frequency cepstrum coefficients (MFCC), spectrogram image features based on short-time fourier transforms (STFT), and wavelet transforms. Since the expression of environmental sounds is theoretically infinite, it is not easy to judge which features are suitable. This is one of the factors that makes it difficult to classify the environmental sounds.

Second, speech and music make meaningful sequences to humans by combining the aforementioned sound units, phonemes and notes. On the other hand, the environmental sound is not. These characteristics of sounds make the models used for classification using sound data different. For example, in the speech recognition field, a hidden markov model (HMM) is mainly used, and recently, a recurrent neural network (RNN) and a long short-term memory (LSTM) belonging to the deep learning framework are used. This is because the speech has a meaningful sequence along the time axis. However, the environmental sound, to which the sounds of rotating machines belong, usually has no meaningful sequences along the time axis. Therefore, there are some studies that deal with environmental sound classification problems in deep learning framework using a convolutional neural network (CNN) instead of sequential models such as the RNN or the LSTM.

Sound Classification with CNN

Deep learning has been successful in many areas. In particular, a CNN, a type of deep learning model in the field of computer vision, has outperformed conventional methods in the classification problem. There have been several efforts to apply deep learning to classification problem with sound data. For environmental sounds, The CNN is often applied instead of sequential models such as RNN or LSTM in general. Some studies have made the sound as two-dimensional data like an image and applied it to CNNs for environmental sound classification[10, 11, 12, 13, 14]. They usually apply STFT to the sound data to construct two-dimensional data, which have time axis and frequency axis.

The above studies have extracted features from sound data and applied it to CNNs, while some studies have applied raw sound data in the time domain directly to CNNs[15, 16, 17, 18]. The CNN in [16], which is called SoundNet by the authors

of [16], uses raw sound data with unlabeled video data and outperform other CNN model in [11] for same sound dataset. This shows the possibility of end-to-end learning on classification of environmental sounds.

1.2 Contributions of This Thesis

Training CNN with Few Sound Measurement Data

In industrial inspection, it is often difficult to obtain many defective samples to train a classifier. In the case of normal samples, too many samples cannot be used for training to balance with the defective samples because the number of the defective samples is small. In addition, recording the sound from the samples is a time-consuming process. As a result, there is a high possibility of learning with a small amount of data in order to learn a classifier for inspection using sound measurement data.

On the other hand, it is advantageous if there is a lot of data in order to obtain high performance using deep learning for classification problem. In this thesis, we propose a method that uses CNN model, which is a type of deep learning, although it has a small amount of data, but performs better than a conventional method in sound classification problem.

End-to-End Learning for Sound Measurement Data

In the classification problem, the approach of conventional methods in machine learning is to first extract engineered features and classify them using the features. However, deep learning seeks to perform feature extraction and classification at once in the model itself. This allows us to input the raw sound data without feature

extraction into the model and get the output we want directly. This is called end-to-end learning.

In order to extract good features in the process of feature extraction, knowledge and understanding of given data are needed. And it is difficult to determine which features to use and a number of parameters in that feature. In this thesis, we use raw sound waveform data as the input of a CNN without feature extraction. Our method is applicable to any rotating machines in many industries, since we do not need to consider the appropriate features for the given data.

1.3 Organization

In Chapter 2, we review some concepts in machine learning that are needed to understand our approach. First, we define the classification problem because our approach starts with considering automated industrial inspection as a classification problem. Second, we introduce an one-dimensional CNN as a model for solving the classification problem. The structure of the CNN and the error backpropagation algorithm, which is an analytical computation method to compute the gradient, are presented together.

We present how to automate industrial inspection in Chapter 3. Since there is only a small amount of data available for industrial inspection, we use transfer learning to solve this. In addition, we propose a layer-wise learning method, called Greedy layer-wise supervised training method, not a method of learning the whole network at a time. This makes it possible to train several layers in the CNN with less data.

Chapter 4 deals with the inspection of electric motors used in drones as an example of industrial inspection. We first describe the process of collecting motor data

and preprocessing method. Through the preprocessed data, the training is carried out through the method presented in Chapter 3. We also introduce the baseline as a conventional classification method in machine learning for evaluation of our algorithm and present some evaluation metrics used in statistics. We compare our algorithm with the baseline using these evaluation metrics and show that our algorithm is superior to the conventional method.

Finally, in Chapter 5, we conclude this thesis by summarizing the research process and the main results.

2

Preliminaries

As mentioned in Chapter 1, we regard industrial inspection using sound measurement data as a classification problem in machine learning. Especially, we solve the classification problem by deep learning which has excellent performance in the field of computer vision recently. Thus, in this chapter, we first define the classification problem and describe the model used in this thesis, a convolutional neural network (CNN).

2.1 Classification Problem

A classification problem is a kind of supervised learning in machine learning. The goal of the classification problem is to assign n-dimensional input vectors $x \in \mathbb{R}^n$ to elements of finite class set $y \in \{1, \cdots, K\}$ where K is the number of classes[19]. This can be thought of as dividing the input space into as many decision regions as the number of classes. The boundary that divides the input space is called the decision boundary.

From another point of view, the classification problem when $y = f(x)$ is to estimate the function f from a given training dataset, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ where N is the number of the training data. We then use the estimated function $\tilde{f}$ to assign the appropriate classes y for new input vector x. What is important in the classification problem is not only to correctly map the input x and output y in the training dataset, but also to predict the output y correctly for the new input vectors x that is not used in training. This is usually called generalization.

To obtain the estimated function $\tilde{f}$, we usually use a specific model $\tilde{f}(x;\theta)$ where θ is model parameters. Then, given the labeled training dataset $\mathcal{D}$, we can estimate the model parameters θ from solving the optimization problem as follows:

$$\min_{\theta} J, \qquad J = \sum_{i=1}^{N} \mathcal{E}(\tilde{f}(x_i;\theta), y_i) \tag{2.1.1}$$

where $\mathcal{E}$ represents a error function that measures the difference between two inputs. Furthermore, we use the categorical cross-entropy function as the error function.

2.2 Convolutional Neural Network (CNN)

In recent years, CNNs have outperformed other methods of classification problem in the field of computer vision[20]. Generally, two-dimensional CNNs are used for image data. In the case of one-dimensional data such as sound data, one-dimensional CNNs can be applied. We use the raw waveform of the sound data as input to our industrial inspection problem. Therefore, we only deal with one-dimensional CNNs.

Figure 2.1 shows a typical form of a one-dimensional CNN. The CNN can be used as a model $\tilde{f}$ to solve the classification problem defined in section 2.1. The CNN consists of repetitions of convolution layers and pooling layers. If input data passes through several convolution and pooling layers, low-dimensional features that

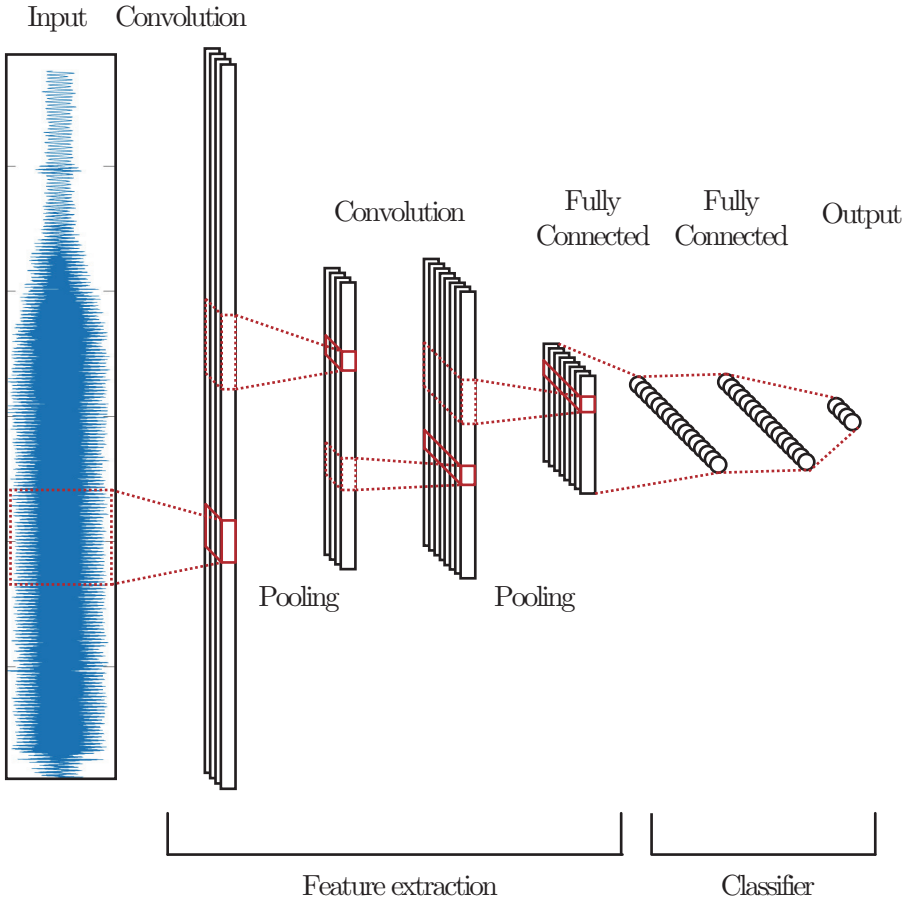


Figure 2.1: Typical structure of an one-dimensional CNN

describes the input data can be obtained. Therefore, the convolution and pooling layers can be regarded as the role of feature extractor. From this extracted features, fully connected layers at the back of the CNN acts as a classifier. The fully connected layers correspond to general artificial neural networks. The last nodes of the CNN exist as many as the number of classes in the classification problem.

2.2.1 Convolution Layer

The convolution layer is an important element in CNNs. Given a n-dimensional input sequence $x \in \mathbb{R}^n$, the convolution layer can be seen as a function of the form

$$h(x) = \sigma(x * w + b), \tag{2.2.2}$$

where σ is an activation function, $w \in \mathbb{R}^p$ and $b \in \mathbb{R}$ denote model parameters with filter size p, and $*$ represents one-dimensional convolution operator. The w is especially called a filter or a kernel in CNN. The i-th element of the output y is computed as follows by the convolution of a input vector x and a kernel vector w:

$$y_i = (x * w)_i \tag{2.2.3}$$

$$= \sum_m x_m \cdot w_{i-m} \tag{2.2.4}$$

$$= \sum_m x_{i-m} \cdot w_m. \tag{2.2.5}$$

In (2.2.4), we can see that when m increases, the element index of the input vector x increases but the element index of the kernel vector w decreases. That is, the kernel w is flipped relative to the input x. Due to the form of this convolution operation, the commutative property of convolution arises. Similarly, the cross-correlation of

a input vector x and a kernel vector w is defined as:

$$y_i = (x \star w)_i \tag{2.2.6}$$

$$= \sum_m x_{i+m} \cdot w_m \tag{2.2.7}$$

where $\star$ denotes the cross-correlation operator. The cross-correlation is an operation similar to convolution operation without flipping the kernel vector w[20].

Generally, an activation function is applied to the output of a convolution layer. This function imposes a nonlinearity on the output of the convolution layer. We use the Rectified Linear Units (ReLU)[21], as an activation function for an input vector x, and the ReLU activation function is defined as:

$$h(x) = \max(0, x). \tag{2.2.8}$$

Furthermore, the softmax activation function is usually applied to the last layer of a CNN in the classification problem. For the input vector x and the output vector y, the softmax function is defined as:

$$y_i = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}} \tag{2.2.9}$$

where the subscripts i and j represent the element index of each vector, and N denotes the number of classes. The y_i obtained from the softmax activation function is considered as the prediction probability for the i-th class. If y_i is the largest, the classifier classifies the corresponding input vector as the i-th class.

In particular, if the dimension of the input vector is large, we may perform subsampling on the input by performing the convolution operation with stride s as

follows:

$$y_i = (x * w)_i \tag{2.2.10}$$

$$= \sum_m x_m \cdot w_{si-m} \tag{2.2.11}$$

$$= \sum_m x_{si-m} \cdot w_m. \tag{2.2.12}$$

where a typical convolution operation is the same as when stride s is one.

2.2.2 Pooling Layer

In CNNs, the output vector of the convolution layer generally passes through the pooling layer as an input. That is, the convolution layer is followed by the pooling layer. The pooling layer divides the input vector into several regions and outputs representative values in each region. When dividing into multiple regions, the input vector is usually divided by non-overlapping filters of a specific size. Since only the representative values are output, the pooling layer has the effect of subsampling the input. It is essential in CNNs because it reduces the amount of computation.

For the representative value in pooling, the most commonly used value is the maximum value and the average value for the input vector. The pooling layer, which uses the maximum value as the representative value, is usually called the max pooling layer, and the pooling layer that uses the average value as the representative value is called the average pooling layer.

2.2.3 Error Backpropagation

To solve the optimization problem in (2.1.1), we can generally use the gradient descent method. Here, it is necessary to obtain the gradient of objective function J with respect to model parameter $\theta = \{w_1, w_2, \cdots, w_L, b_1, b_2, \cdots, b_L\}$, where L is

the number of layers.

The error backpropagation is an analytic method to obtain the gradient of the objective function J. To find the gradient of J with respect to the model parameters, we need to find the gradient of the error function $\mathcal{E}$ with respect to the model parameters θ.

Therefore, we propose a procedure to find the gradient of $\mathcal{E}$ with respect to model parameters of the l-th convolution layer, w_l and b_l, as an example of the error backpropagation algorithm. First, we represent the l-th convolution layer as follows:

$$a_l = h_{l-1} * w_l + b_l \tag{2.2.13}$$

$$h_l = \sigma(a_l) \tag{2.2.14}$$

Using the chain rule, the gradient of $\mathcal{E}$ with respect to w is derived as:

$$\frac{\partial \mathcal{E}}{\partial w_{l,i}} = \sum_k \frac{\partial \mathcal{E}}{\partial a_{l,k}} \cdot \frac{\partial a_{l,k}}{\partial w_{l,i}} \tag{2.2.15}$$

$$= \sum_k \delta_{l,k} \cdot h_{l-1,k-i} \tag{2.2.16}$$

$$= \delta_{l,i} * h_{l-1,-i} \tag{2.2.17}$$

$$= h_{l-1,i} \star \delta_{l,i} \tag{2.2.18}$$

where the subscripts denotes the element index of each vector, and δ_l is defined as $\delta_l \triangleq \frac{\partial \mathcal{E}}{\partial a_l}$. Similarly, we can also derive the gradient of $\mathcal{E}$ with respect to b using the chain rule as follows:

$$\frac{\partial \mathcal{E}}{\partial b_l} = \sum_k \frac{\partial \mathcal{E}}{\partial a_{l,k}} \cdot \frac{\partial a_{l,k}}{\partial b_l} \tag{2.2.19}$$

$$= \sum_k \delta_{l,k} \tag{2.2.20}$$

For δ_l, the following relation can be obtained by using the chain rule and δ_{l+1}:

$$\delta_{l,i} = \sum_k \frac{\partial \mathcal{E}}{\partial a_{l+1,k}} \cdot \frac{\partial a_{l+1,k}}{\partial a_{l,i}} \tag{2.2.21}$$

$$= \sum_k \delta_{l+1,k} \cdot w_{l+1,k-i} \cdot \sigma^{'}(a_{l,i}) \tag{2.2.22}$$

$$= \sigma^{'}(a_{l,i}) \cdot w_{l+1,i} \star \delta_{l+1,i} \tag{2.2.23}$$

where $\sigma^{'}$ represents the derivative of the activation function σ. To obtain δ of each layer, we need δ of the last layer first. Using (2.2.23), we can get δ of the previous layer in reverse order. This calculation method caused the term backpropagation. Gradients in a pooling layer and a fully connected layer can be obtained in a similar manner.

3

CNN for Industrial Inspection

Recently, deep learning has shown excellent performance in the field of computer vision, but it is possible with many labeled data. However, in the case of sound data, it is difficult to collect more data than the image. In particular, it is difficult to collect the samples necessary to learn the classifier for the sounds of rotating machines, such as engines or motors, used in industries. Therefore, industrial inspection using sound measurement data is usually carried out with limited data.

In this chapter, we present a deep learning architecture and method which can perform better even if less sound data is given. We also propose a training method that can achieve better performance in the CNN using transfer learning.

3.1 CNN Architecture for Transfer Learning

3.1.1 Transfer Learning

Transfer learning is a technique that can improve classification performance when there is less labeled data. In general, gathering and labeling many data all together

is a cumbersome and tedious task. Therefore, transfer learning can be an effective way to improve performance while reducing the effort to obtain a lot of data.

Transfer learning aims to transfer knowledge of source tasks to a target task[22]. In other words, transfer learning is to initialize the CNN as the target with the pre-trained filters as source domain dataset $\mathcal{D}_S = \{x_i, y_i\}_{i=1}^{N_S}$, where N_S is the number of source domain data. Then, the initialized CNN is generally learned by the following two methods using the target domain dataset $\mathcal{D}_T = \{x_i, y_i\}_{i=1}^{N_T}$, where N_T is the number of target domain data. The first method is fine-tuning of the entire CNN layers. This method can be effective if the given target domain dataset $\mathcal{D}_\mathcal{T}$ is large. The second method is to fix the filters of some CNN layers (frozen network) and fine-tuning the rest. This method can be effective when the target domain dataset $\mathcal{D}_\mathcal{T}$ is small. In [23], experimental results show that the performance of fixing the filters of several layers of a CNN is high when the number of target domain data is small.

3.1.2 CNN Architecture

We use SoundNet model presented in [16] as a source model for using transfer learning. In transfer learning, the source model should learn the general features of the data. In order to learn the general features of data, it is advantageous to have more labels and more data in the given data.

SoundNet model is an one-dimensional CNN model trained from numerous unlabeled videos. The dataset used in SoundNet model consists of about 2 million videos. These videos are downloaded online and contain everyday contents. In addition, this dataset includes videos about the engines, which are also related to the sound of the rotating machines we are interested in. The authors for SoundNet train the CNN by extracting sounds from video data. Here, the frame images of the video

Table 3.1: Our CNN model configuration

Group name	**Layer**	**Filter size**	**# of filters**	**Stride**
	Input signal			
Layer group 1	Convolution	64	16	2
	Max Pooling	8	-	-
Layer group 2	Convolution	32	32	2
	Max Pooling	8	-	-
Layer group 3	Convolution	16	64	2
Layer group 4	Convolution	8	128	2
Layer group 5	Convolution	4	256	2
	Max Pooling	4	-	-
Classifier	Convolution	1	# of classes	-
	Global Average Pooling	-	-	-
	Softmax	-	-	-

data serve as labels of the extracted sound. From the above, Soundnet is a model that has many labels and is trained from a lot of data. Therefore, it is suitable for use as a source model in transfer learning.

SoundNet model has an 8-layer and 5-layer version, and we use part of the 8-layer version, which is up to the 5th layer, as our CNN model. The reason for this is that it gave the best performance when various sound data was classified using the features extracted from the 5th layer of the 8-layer version model in [16]. The CNN model we use is shown in Table 3.1.

The layer groups 1 through 5 in Table 3.1 are the same as the architecture of SoundNet. It consists of repetitions of one-dimensional convolution layers and one-dimensional max pooling layers. The meaning of one dimensional here is that the convolution and the pooling operation are performed along the time axis. Also, a batch normalization[24] layer and a ReLU[21] activation layer are used after all

convolution layers. In Table 3.1, these are omitted.

For the classifier part in Table 3.1, we use a new structure that is not used in SoundNet. The first convolution layer of the classifier in Table 3.1 has a filter size of 1. The convolution layer with filter size of 1 is used in GoogLeNet[25]'s Inception module. This layer maps the number of filters to the number of classes in a given problem. The values of each filter are then averaged in the global average pooling layer. Then, the probability per class is obtained with the softmax activation function. The classifier in Table 3.1 reduces the number of parameters in the model more than the fully connected layers used at the end of conventional CNNs. Thus this new structure reduces the complexity of the model and prevents overfitting.

3.2 Greedy Layer-wise Supervised Training Method

The industrial inspection problems we deal with are usually difficult to obtain many labeled samples. When applying CNNs to computer vision problems, millions of data are often used. In the case of image data, a lot of data can be quickly obtained through cameras, and many photos are shared online by many people using social network service (SNS) recently. However, in industrial inspection using sound measurement data, it is difficult to obtain a lot of defective samples and it takes a lot of time to record the sound of the samples. Therefore, the number of data that can be used in the industrial inspection problem using sound measurement data is usually small.

The performance of classification using a CNN is likely to be better with more data and deeper layers. The deeper the layer of the CNN, the greater the number of parameters to learn, which greatly increases the complexity of the model. Therefore, the possibility of overfitting increases. There is less data available for our industrial

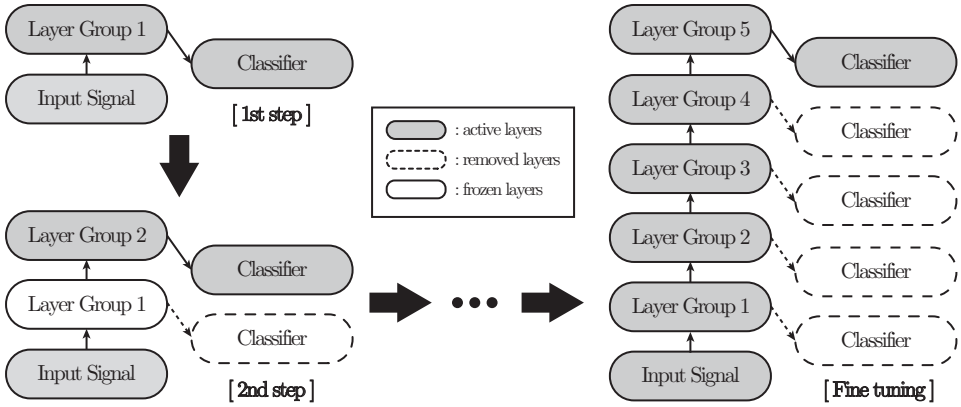


Figure 3.1: Illustration for Greedy layer-wise supervised training method

inspection problems, so it is difficult to use many layers on the CNN. However, we can use more layers by applying transfer learning from well-trained CNN structures. In addition, we present a learning method that allows more layers to be used while applying transfer learning.

Generally, the more layers are stacked, the more difficult it is to learn because of vanishing gradient problem. In [26], the authors enabled the learning by reducing the influence of the vanishing gradient problem by layer-wise training method. Similarly, we propose Greedy layer-wise supervised training method that can be used in classification problems. This training method learn the CNN model in a supervised manner.

Greedy layer-wise supervised training method is the same as the method presented in [27]. However, unlike [27], we use it with transfer learning and apply it to the sound data, not the image. Greedy layer-wise supervised training method shown in Figure 3.1 is as follows:

- Grouping layers: First, we set up the entire CNN structure. Then, as in Table 3.1, the CNN structure is divided into several Layer groups. At this stage, it is not necessary to include only one convolution layer in one group.

- 1st step: As shown in Figure 3.1, we connect the classifier in Table 3.1 to the layer group 1 and learn the CNN using the error backpropagation algorithm. Here, the filters of layer group 1 are initialized to the filters of 8-layer version of SoundNet.

- 2nd step: After learning the first step, remove the classifier and connect the next layer group and the classifier in Table 3.1 to the CNN structure of the previous step. Here, the filters of layer group 1 is fixed using the filters learned in the previous step. The filters of layer group 2 is initialized to the filters of 8-layer version of SoundNet. Then learn the entire CNN structure. In this case, only the part of layer group 2 is learned. In the following layer groups, layer-wise learning proceeds in the same way.

- Fine tuning: After learning all the layer groups, we initialize each layer group to the filters learned in the last step. Figure 3.1 shows the case of learning up to layer group 5. The entire CNN structure is then trained using the error backpropagation.

4

Experiment for Electric Motor Inspection

The CNN model shown in Chapter 3 can be applied to inspection problems using sound signal of any rotating machines. Because our algorithms are a kind of end-to-end learning method, we can extract features useful for classification in the model itself without a deep understanding of the given data. This characteristic makes it possible to apply a consistent method to all sound measurement data.

Typical examples of rotating machines used in industries are engines and motors. Among them, we use the sound measurement data of electric motors used in drones, and we implement experiments for inspection using these data. We evaluate the performance of our CNN model from these experiments. To do this, we present various model evaluation methods used in statistics and the performance of our model based on them.

4.1 Data Acquisition

The electric motors used in our experiments are the DJI 2312 brushless motor used in DJI's Phantom 3 drones. To measure the sounds as the electric motors run, we construct a sound recording environment as shown in Figure 4.1. T-Motor ESC T60A 400Hz is used as the controller to control the motors and Arduino mega is used as the analog-to-digital converter. The laptop gives the motor operating signal and stores the recorded signal from the connected microphone. Since the industrial inspection is mainly carried out at the factory, the sound data recorded at the DJI factory are applied through the external speaker as noise in order to create a factory environment. To minimize the noise, the motor and microphone are operated within a soundproof box.

The step input is applied using the laptop in order to operate the electric motors. The recording time for each motor is 12 seconds, which is enough time to get all the sounds from the moment when the electric motor starts rotating to the moment when the rotation stops. The sampling rate of recorded sounds is 48,000Hz. Recorded sounds have single channel.

4.2 Problem Definition and Data Preprocessing

We regard this electric motor inspection problem as a classification problem in machine learning. This is a binary classification problem with two classes, normal and defective, and the CNN in Chapter 3 is used as our model. We use 250 normal motors and 199 defective motors in this experiments. The data labeling of the electric motors was carried out by well trained specialists. The defective motors have different degrees of defects. Therefore, in some motors, the defects are not serious

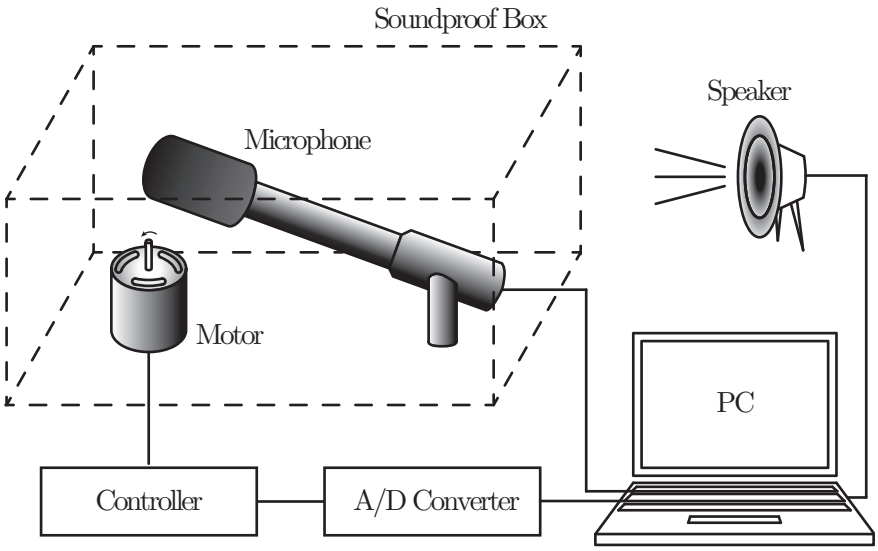
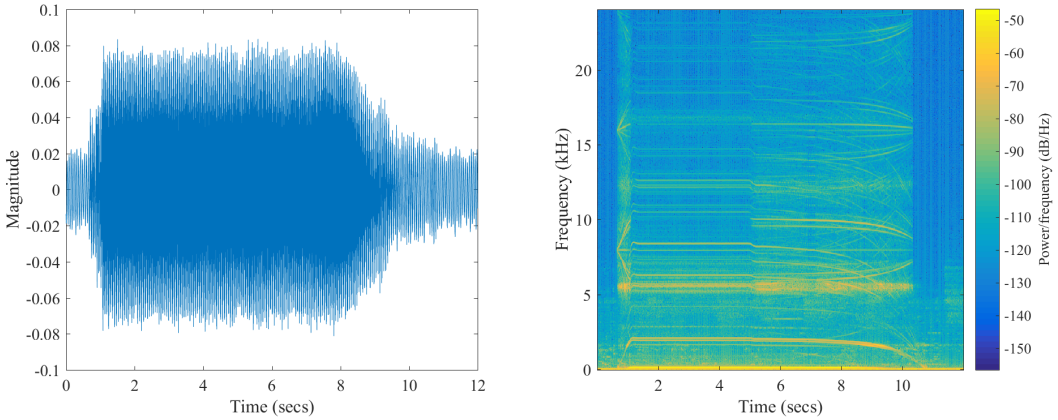


Figure 4.1: Schematic diagram of data acquisition system for electric motor inspection

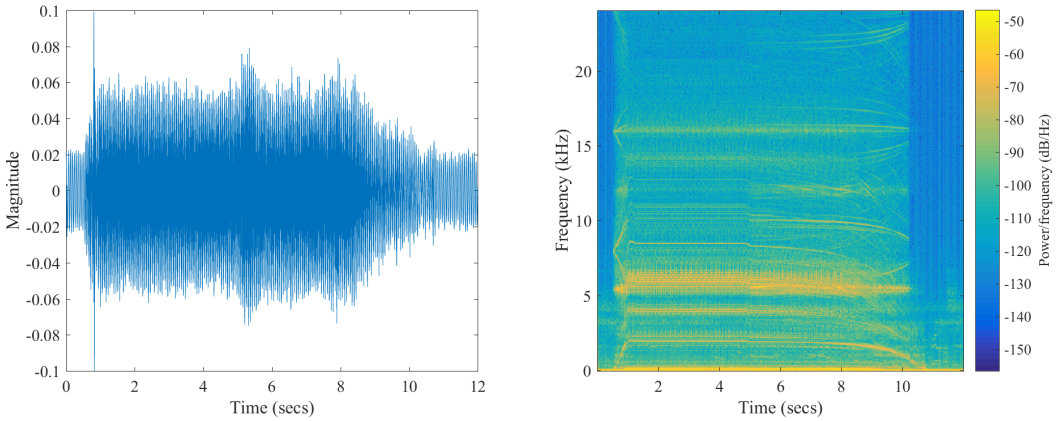
and it is sometimes difficult for a person to distinguish defects from sound.

Figure 4.2 shows the raw waveform signal and the spectrogram of a normal motor sound and a defective motor sound. For obtaining the spectrogram from motor sounds, we use a hamming window of length 1,024 with half overlapping samples. In the spectrogram, we can see that there is a low power per frequency section on the front and back of the sound. This part is removed later because there is no motor sound.

The previously obtained sound data is a very high dimensional vector of 576,000, which is calculated by 48,000Hz×12sec. If the input dimension is large, it requires a lot of data because of the curse of dimensionality in order to classify it. Therefore, preprocessing is required to reduce the dimension of the data while minimizing the loss of information. The preprocessing process in this experiment is as follows:



(a) Normal motor



(b) Defective motor

Figure 4.2: An example of recorded motor sounds as raw waveform and spectrogram

- Resampling signal: When learning the CNN using transfer learning, we resample the data at 22,050Hz in order to make it equal to the sampling rate used in the source domain. We use the polyphase filtering method to resample the data.

- Normalizing: To normalize the resampled data, we subtract the mean from the original data and divide it by the standard deviation. This is to preserve the distribution of power of the motor sound. After normalization, we multiply the data by 100 to adjust the amplitude.

- Removing less important part: There is little meaningful sound at the front and back of the motor sound, which can be found in the spectrogram of Figure 4.2. Therefore, we remove 1 second from the front of the sound and 2 seconds from the back. Finally, we use 9 seconds of motor data.

It should be noted that data augmentation was not performed here. In general, we can use data cropping techniques, which are mostly used in image data, to increase the data for sound. However, for a defective sound, it cannot be guaranteed that the entire data section is a defective sound. That is, there may be a normal sound in the data that has been cropped from the defective sound. Therefore, we do not perform data augmentation because labeling class of cropped data is ambiguous.

Finally, since we have a small amount of electric motor data, it is difficult to learn the CNN from scratch. To solve this problem, we use transfer learning using Greedy layer-wise supervised training method presented in Chapteer 3 by inputting preprocessed motor sound data. The goal is to learn CNN's filters from the training data so that the motor sound data not used for learning can be correctly classified

into two classes. From this we have to filter out the defective motors.

4.3 Evaluation Procedure

In order to evaluate the model presented in Chapter 3, comparative objects are needed. We present a model as a baseline and evaluate the performance of our model against the baseline model by means of statistically significant measures.

4.3.1 A Baseline as Conventional Methods

For the classification problem, conventional methods rather than deep learning are to extract features from given data and learn a classifier from the features. We construct a baseline as a conventional method that uses continuous wavelet transform (CWT) as a feature and support vector machine (SVM) as a classifier.

CWT is a type of wavelet transform that is a mathematical tool for mapping signals in the time domain to other domain. CWT of a signal $x(t)$ is obtained by a convolution operation with a specific complex conjugate function and is expressed as[28]

$$cwt(s,\tau) = \frac{1}{\sqrt{s}} \int x(t)\psi^* \left(\frac{t-\tau}{s} \right) dt \tag{4.3.1}$$

where the symbols $s \in \mathbb{R}^+$ and $\tau \in \mathbb{R}$ denotes scale and translation parameters, respectively. ψ^* is continuous function and complex conjugate of wavelet function ψ. It is known to be useful for non-stationary signals and has been widely used for fault diagnosis of rotary machines[29].

To extract features using CWT, Daubechies 8-tap (db8) wavelet function is used as wavelet function ψ. The scaling parameter ranges from 8 to 128 as in [5]. For

motor sounds, we use the sampling rate of 22,050Hz as our case. We take root-mean-square along the time axis for the signal passed through CWT and obtain a 121-dimensional vector for each motor sound. We use these vectors as the features and train a linear SVM using them.

4.3.2 5-fold Cross Validation

What is important to the classification problem is whether the trained classifier can correctly classify data that is not used for training, which is called generalization. In other words, it is important to train not to overfit the training data. Generally, we use some of our data as a training set, which is a dataset used in training, and the rest as a test set, which is a dataset used in test. Then we train the classifier only with the training set and apply it to the test set to see if the generalization is good. From the classification results for the test set we can evaluate the performance of the classifier. However, the final classification performance depends on how the data is divided into a training set and a test set. Therefore, we use k-fold cross validation as a way to properly evaluate the performance of the model while minimizing this effect.

k-fold cross validation is a method of dividing the entire data into k folds, using one fold as a test set and the remaining data as a training set. After repeating this k times, the results of each trial are summarized and presented as the final performance of the model. Generally, when k increases, the amount of computation increases, but there is an advantage that much data can be used for learning.

In our experiment, 5-fold cross validation is used and its configuration is shown in Figure 4.3. The motor sound data are divided into five folds with the same configuration as Figure 4.3 and five experiments are carried out. For each experiment, we use the gray fold in Figure 4.3 as a test set and the remaining folds as a training

Experiment 1 :	Normal: 50 Defective: 40	Normal: 50 Defective: 40	Normal: 50 Defective: 40	Normal: 50 Defective: 40	Normal: 50 Defective: 39
Experiment 2 :	Normal: 50 Defective: 40	Normal: 50 Defective: 40	Normal: 50 Defective: 40	Normal: 50 Defective: 40	Normal: 50 Defective: 39
Experiment 3 :	Normal: 50 Defective: 40	Normal: 50 Defective: 40	Normal: 50 Defective: 40	Normal: 50 Defective: 40	Normal: 50 Defective: 39
Experiment 4 :	Normal: 50 Defective: 40	Normal: 50 Defective: 40	Normal: 50 Defective: 40	Normal: 50 Defective: 40	Normal: 50 Defective: 39
Experiment 5 :	Normal: 50 Defective: 40	Normal: 50 Defective: 40	Normal: 50 Defective: 40	Normal: 50 Defective: 40	Normal: 50 Defective: 39

Figure 4.3: Configuration of motor data for 5-fold cross validation

set. Finally, after learning the classifier, the area under the ROC curve (AUC), to be covered next section, of our CNN model are presented as the final performance of our model.

4.3.3 ROC Curve and AUC

A receiver operating characteristics (ROC) curve is an effective technique for visualizing the performance of a binary classifier[30]. Generally, we set positive and negative samples of the binary classification problem. In this case, the ROC curve is a graph drawn by plotting true positive rate (TPR) against false positive rate (FPR) by changing the threshold of the classifier. TPR means the proportion of samples correctly classified as positive among positive samples. FPR is the ratio of samples classified as negative among positive samples.

We can obtain the area under the ROC curve, which is called AUC. AUC can be used as an evaluation metric of the classifier. The larger AUC of the classifier, the better the classifier. In particular, AUC of 1 means a perfect classifier.

In our electric motor inspection problem, we see the normal motors as positive samples and the defective motors as negative samples. To obtain an ROC curve for a classifier, we collect all TPR and FPR obtained by changing the threshold of the classifier from five experiments using 5-fold cross validation. We draw one ROC curve from the collected TPR and FPR. From this one ROC curve, the classifier gets one AUC.

4.4 Experimental Results

Our experiment is implemented using Keras[31], a python-based deep learning library, with Theano[32] backend. We use the Adam[33] optimizer and the learning rate is 0.0001 for all experiments. We use the batch size as 32 in optimization. Other model parameters, such as momentum in batch normalization[24], use the default value of Keras. We use the glorot initialization[34] when performing random initialization on filters of CNNs. All implementations run on an Intel (R) Core (TM) i7-6700 CPU @ 3.40GHz and use an NVIDIA Geforce TITAN X (Pascal) GPU to accelerate calculations.

4.4.1 Effect of Greedy Layer-wise Supervised Training Method

To investigate the effect of Greedy layer-wise supervised training method (greedy training) in transfer learning, we compare greedy training with the other three learning methods. One method is to randomly initialize filters of each layer and learn from scratch (random initialization). The other two methods are used for transfer learning: fine-tuning the entire network (fine-tuning), fixing filters of some front layers and fine-tuning filters of remaining layers (frozen network).

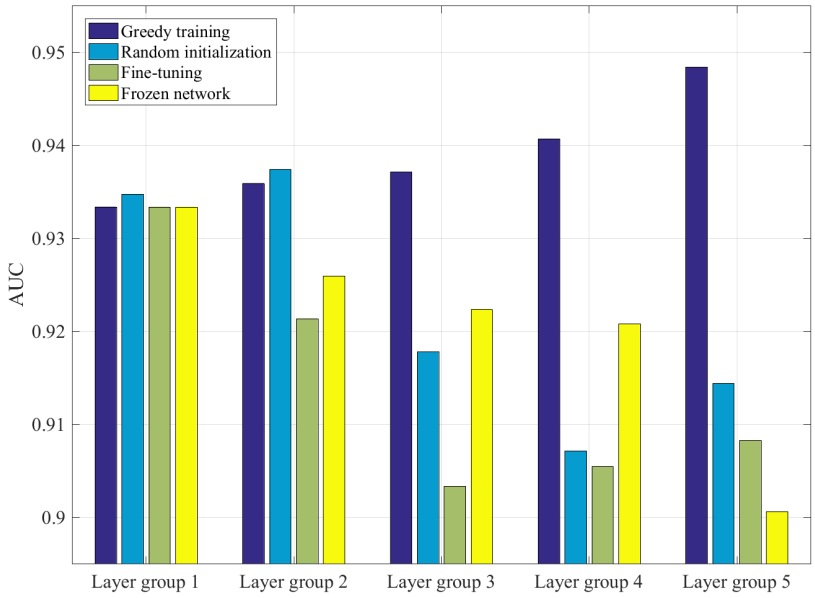


Figure 4.4: Area under ROC curve (AUC) for various training methods

The AUCs of the classifiers learned by these four methods are shown in Figure 4.4. The horizontal axis of Figure 4.4 represents the last layer group used in the training. For example, in the case of random initialization in Figure 4.4, the AUC in the layer group 3 is the result of learning from scratch using the layer group 1 to 3 and the classifier part in Table 3.1. In the frozen network shown in Figure 4.4, the AUC of the layer group 3 is the result of learning only the layer group 3 and the classifier part in Table 3.1 while fixing the filters up to layer group 2. The training of the frozen network proceeds after initializing the filters with the filters of 8-layer version of SoundNet.

Figure 4.4 shows that as the layer becomes deeper, the AUC of greedy training

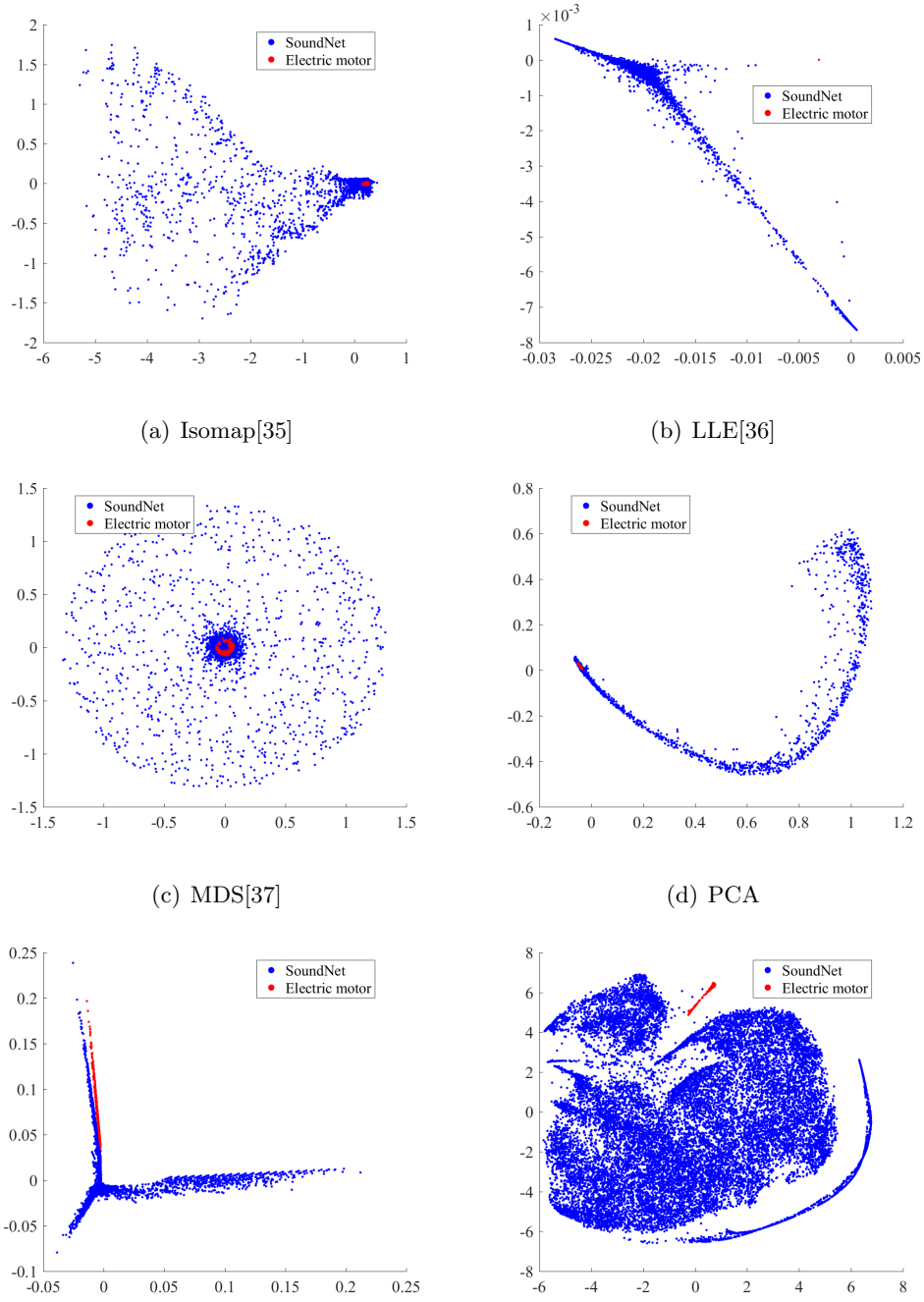
Table 4.1: Number of parameters of our CNN model

Group name	# of parameters
Layer group 1	1,072
Layer group 2	16,480
Layer group 3	32,960
Layer group 4	65,920
Layer group 5	131,840
Classifier	514
Total	248,786

increases, while the other methods tend to decrease the AUC as the layer becomes deeper. In general, as shown in Table 4.1, the deeper the layer in a CNN, the more parameters to learn. If there are a large number of parameters to be learned, there should be a lot of data to be used for learning in order to prevent overfitting. But the dataset given in our electric motor inspection problem is small. Therefore, random initialization degrades performance when using many layers.

In Figure 4.4, it is noteworthy that the result of transfer learning, which are useful when there is less training data, is not better than that of random initialization. In the layer group 3 and 4, the AUCs of frozen network, which is a kind of transfer learning, are higher than that of random initialization but lower than that of random initialization in the layer group 2.

To investigate the cause of performance degradation in transfer learning, we perform two-dimensional feature embedding with a source domain dataset and a target domain dataset through various algorithms and visualize it as Figure 4.5. The source domain dataset denotes the data used for SoundNet[16] learning and



(a) Isomap[35]

(b) LLE[36]

(c) MDS[37]

(d) PCA

(e) Spectral Embedding[38]

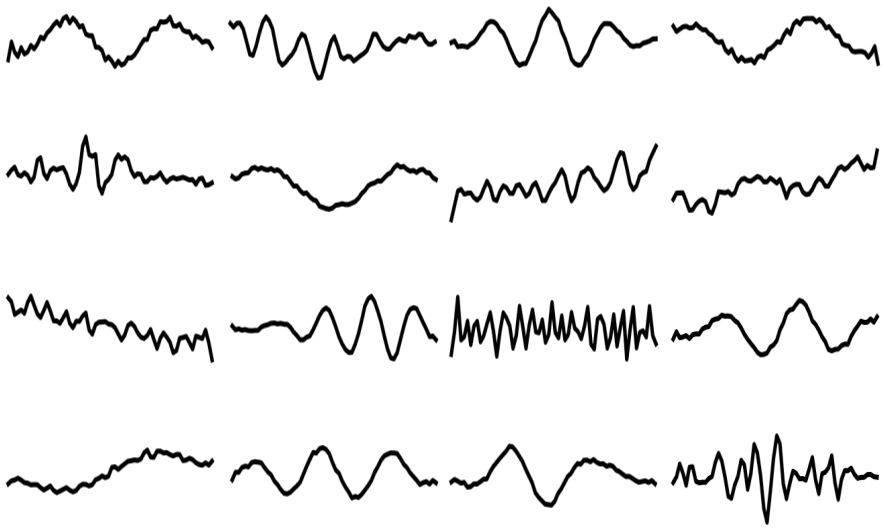
(f) t-SNE[39]

Figure 4.5: Two-dimensional feature embedding using various embedding algorithms

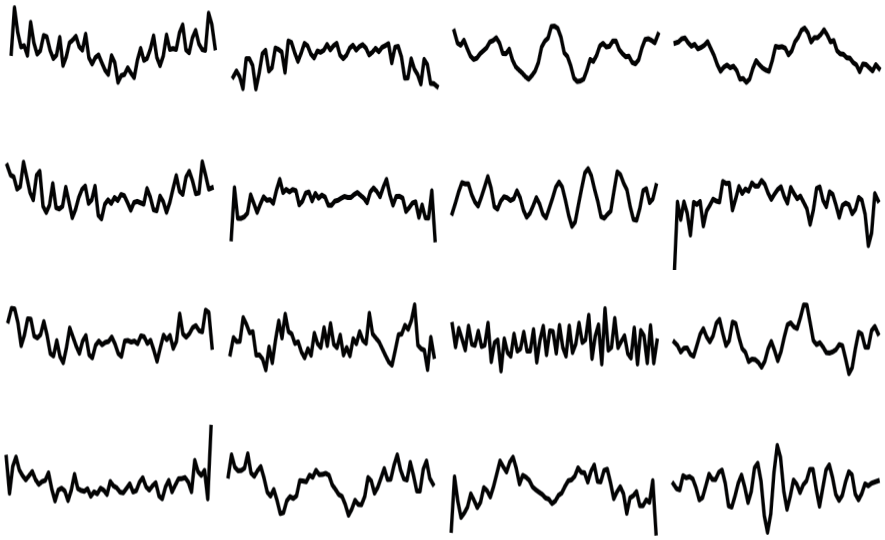
the target domain dataset represents our electric motor data. In the case of the source domain, a subset of size 20,000 is randomly extracted from the entire source domain data to perform feature embedding. Feature embedding is performed with features from the 5th layer of SoundNet for both datasets. In most of the results in Figure 4.5, the target domain is slightly overlapping or entirely apart from a portion of the source domain. This means that the target domain is associated with only a small fraction of the source domain, which is less relevant when compared to the entire source domain data. Therefore, the performance of the transfer learning may be low because the learned SoundNet filters are not suitable for the target domain.

Nevertheless, greedy training in Figure 4.4 shows higher performance as the layer becomes deeper than other methods. To investigate the effect of greedy training, we visualize the filters of the first layer of SoundNet and our CNN model as Figure 4.6. Figure 4.6(b) shows the filters from greedy training after initializing the filter of the first layer to Figure 4.6(a). The filters in the same position in Figure 4.6(a) and 4.6(b) are the corresponding filters. That is, the first filter in Figure 4.6(b) is the newly learned filter from the first filter in Figure 4.6(a). In Figure 4.6(a), filters of various frequencies are trained while in Figure 4.6(b), filters with higher frequencies are trained. These results show that greedy training adapts the learned filters for the source domain to our electric motor data.

Based on the above, the effect of greedy training in transfer learning is to improve the performance by adapting the learned filters to the target domain even if the source and target domains are not highly related. In addition, filters pre-trained from the source domain can help prevent overfitting and deepen layers when learning with little target domain data.



(a) 8-layer version SoundNet



(b) After greedy layer-wise training

Figure 4.6: Filter visualization on the first layer

Table 4.2: Performance comparison with AUC

Method	AUC
Greedy training	**0.9484**
CWT+SVM	0.9039

4.4.2 Comparison of Results

In this section we compare greedy training with the baseline presented in section 4.3.1 as a conventional method for motor inspection. In the case of greedy training, we use the result of learning up to the layer group 5, which is the best performance in the previous section. The baseline uses CWT as a feature and SVM as a classifier. In both cases, 5-fold cross validation is applied, and the data configuration in each fold is the same. Table 4.2 shows the AUCs for the two cases. In Table 4.2, the AUC of greedy training is larger than the AUC of the conventional method using CWT and SVM. This means that the classification performance of greedy training is better than the conventional method.

In industrial inspection it is important that the defective products are not included in the final products. Therefore, the learned classifier should have high true negative rate (TNR), which is the percentage of motors classified as defective among the defective motors in our electric motor inspection problem, although it is also important that AUC is high. Also, if TPR is high with high TNR of classifier, many normal products may be included in final products without including defective products. In other words, This means that productivity is improved with a low defect rate.

Figure 4.7 shows TPR against TNR, which is obtained by varying the threshold of the classifier as it is when drawing an ROC curve. Generally, as TNR increases,

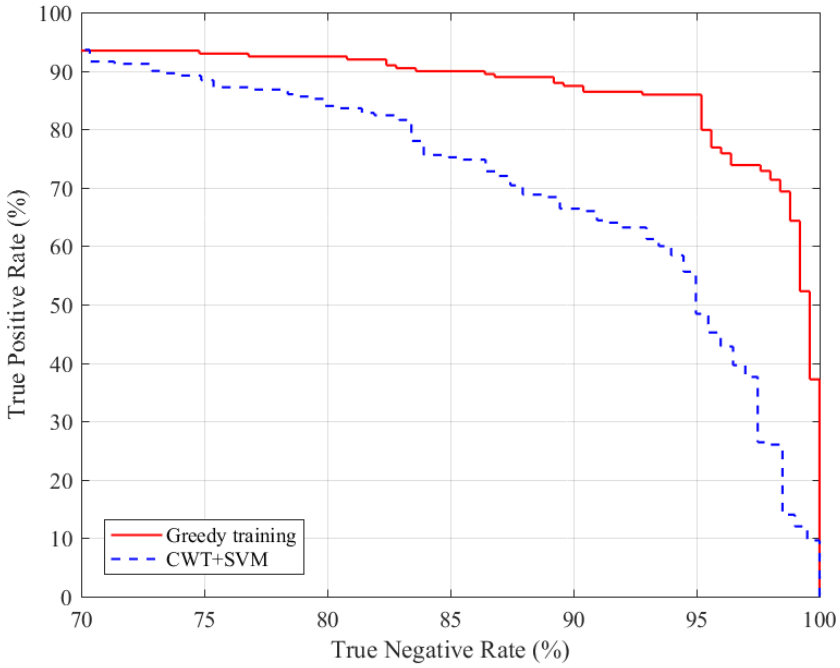


Figure 4.7: True positive rate against true negative rate

TPR decreases. In Figure 4.7, TPR decreases rapidly with increasing TNR for the baseline. On the other hand, the greedy training maintains high TPR until the TNR reaches about 95%. These results indicate that greedy training has higher productivity with higher detection rate than that using CWT and SVM. From this, it can be seen that transfer learning using greedy layer-wise supervised training is more suitable for industrial inspection than the conventional method.

5

Conclusion

In this thesis, we have discussed a method for automating industrial inspection using sound measurement data. For rotating machines, such as engines and motors, it is adequate to check for abnormalities by examining the working sound than by visually inspecting them. However, if a person directly examines the sound measurement data, it takes a long time to hear all the sounds and there is a possibility of mistakes due to their fatigue. Therefore, it is necessary to automate industrial inspection using sound measurement data.

We consider industrial inspection as a classification problem in machine learning to automate industrial inspection using sound measurement data. Furthermore, a CNN is introduced to solve the classification problem. In general, the CNN, a type of deep learning, requires a lot of data to learn. However, it is usually difficult to collect a lot of data for training in industrial inspection because it is difficult to obtain a large number of defective samples in industrial inspection problems.

We use transfer learning to learn with less data. However, since the source and target domain are not very relevant, it is difficult to achieve good performance with

general transfer learning methods. To overcome this problem, we propose Greedy layer-wise supervised training method. By using greedy training, we can accumulate many layers and achieve high performance with the help of pre-trained source model. Especially when the TNR is high, it has higher TPR than the conventional method. This is an important characteristic of the classifier in industrial inspection and means that the defect rate is low and productivity is high.

Furthermore, the algorithm we proposed is a kind of end-to-end learning. In the conventional classification method in machine learning, features are extracted from given data and classified. In order to extract proper features, it is necessary to understand the given data and it is difficult to determine optimal parameters of feature extractor. This means that if the object of industrial inspection is changed, a new feature extraction method and an effort to find the optimal parameters are again necessary. However, our method uses raw waveform of given sound data as an input and learns features corresponding to the given data on the CNN. This means that whatever type of sound measurement data we apply to our method, we can proceed in a consistent way and do not need a deep understanding of the given data.

Finally, our algorithm is a deep learning based algorithm. Therefore, the more data used for training, the better the performance. If we can collect more samples for use in a real industry, our algorithm is expected to perform better than now.

Bibliography

[1] D. Gerhard. *Audio signal classification: History and current techniques*. Citeseer, 2003.

[2] K. Shibata, A. Takahashi, and T. Shirai. Fault diagnosis of rotating machinery through visualisation of sound signals. *Mechanical Systems and Signal Processing*, 14(2):229–241, 2000.

[3] W. Li and C. K. Mechefske. Detection of induction motor faults: a comparison of stator current, vibration and acoustic methods. *Journal of vibration and Control*, 12(2):165–188, 2006.

[4] U. Benko, J. Petrovcic, D. Jurivcic, J. Tavcar, and J. Rejec. An approach to fault diagnosis of vacuum cleaner motors based on sound analysis. *Mechanical Systems and Signal Processing*, 19(2):427–445, 2005.

[5] P Chattopadhyay and P Konar. Feature extraction using wavelet transform for multi-class fault detection of induction motor. *Journal of The Institution of Engineers (India): Series B*, 95(1):73–81, 2014.

[6] E. Germen, M. Başaran, and M. Fidan. Sound based induction motor fault diagnosis using kohonen self-organizing map. *Mechanical Systems and Signal Processing*, 46(1):45–58, 2014.

[7] J. Wu and C. Liu. Investigation of engine fault diagnosis using discrete wavelet transform and neural network. *Expert Systems with Applications*, 35(3):1200–1213, 2008.

[8] J. Wu, E. Chang, S. Liao, J. Kuo, and C. Huang. Fault classification of a scooter engine platform using wavelet transform and artificial neural network. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 1, pages 18–20. Citeseer, 2009.

[9] F. Alías, J. C. Socoró, and X. Sevillano. A review of physical and perceptual feature extraction techniques for speech, music and environmental sounds. *Applied Sciences*, 6(5):143, 2016.

[10] H. Zhang, I. McLoughlin, and Y. Song. Robust sound event recognition using convolutional neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 559–563. IEEE, 2015.

[11] K. J. Piczak. Environmental sound classification with convolutional neural networks. In *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on*, pages 1–6. IEEE, 2015.

[12] H. Phan, L. Hertel, M. Maass, and A. Mertins. Robust audio event recognition with 1-max pooling convolutional neural networks. *arXiv preprint arXiv:1604.06338*, 2016.

[13] J. Salamon and J. P. Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *arXiv preprint arXiv:1608.04363*, 2016.

[14] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, et al. CNN architectures for large-scale audio classification. *arXiv preprint arXiv:1609.09430*, 2016.

[15] P. Golik, Z. Tüske, R. Schlüter, and H. Ney. Convolutional neural networks for acoustic modeling of raw time signal in LVCSR. In *INTERSPEECH*, pages 26–30, 2015.

[16] Y. Aytar, C. Vondrick, and A. Torralba. Soundnet: Learning sound representations from unlabeled video. In *Advances in Neural Information Processing Systems*, pages 892–900, 2016.

[17] S. Qu, J. Li, W. Dai, and S. Das. Understanding audio pattern using convolutional neural network from raw waveforms. *arXiv preprint arXiv:1611.09524*, 2016.

[18] W. Dai, C. Dai, S. Qu, J. Li, and S. Das. Very deep convolutional neural networks for raw waveforms. *arXiv preprint arXiv:1610.00087*, 2016.

[19] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[20] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[21] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[22] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[23] D. Soekhoe, P. van der Putten, and A. Plaat. On the impact of data set size in transfer learning using deep neural networks. In *International Symposium on Intelligent Data Analysis*, pages 50–60. Springer, 2016.

[24] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[26] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, et al. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153, 2007.

[27] D. R. Plata, R. Ramos-Pollán, and F. A. Gonzalez. Effective training of convolutional neural networks with small, specialized datasets. *Journal of Intelligent & Fuzzy Systems*, 32(2):1333–1342, 2017.

[28] O. Rioul and M. Vetterli. Wavelets and signal processing. *IEEE signal processing magazine*, 8(4):14–38, 1991.

[29] R. Yan, R. X. Gao, and X. Chen. Wavelets for fault diagnosis of rotary machines: A review with applications. *Signal Processing*, 96:1–15, 2014.

[30] T. Fawcett. An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[31] F. Chollet et al. Keras. https://github.com/fchollet/keras, 2015.

[32] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

[33] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[34] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, volume 9, pages 249–256, 2010.

[35] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[36] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[37] I. Borg and P. Groenen. Modern multidimensional scaling: theory and applications. *Journal of Educational Measurement*, 40(3):277–280, 2003.

[38] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[39] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

국문초록

이 논문에서는 음향 측정 데이터를 이용한 산업용 검사를 자동화 하기 위해 컨벌루션 신경망 모델을 이용한 검사 방법을 제안한다. 우리는 먼저 산업용 검사 문제를 자동화하기 위해 이를 기계 학습에서의 분류 문제로 간주한다. 음향 측정 데이터로 검사할 수 있는 회전하는 기계에 대한 정상, 불량 샘플들의 소리 데이터가 주어졌을 때, 우리는 이로부터 딥러닝의 일종인 컨벌루션 신경망을 이용하는 분류기를 학습한다. 일반적으로 산업용 검사 문제에서는 학습에 필요한 데이터를 대량으로 얻기가 힘들다. 우리는 이러한 학습 데이터의 부족을 극복하기 위해 전이 학습을 사용한다. 추가적으로 전이 학습에서의 성능 향상을 위해 탐욕적 층별 지도 학습 방법을 제시한다. 음향 측정 데이터를 이용한 산업용 검사의 하나의 예로 드론에 사용되는 전기 모터에 대한 검사를 앞서 제시한 방법으로 수행한다. 드론의 음향 측정 데이터가 주어졌을 때 우리의 알고리즘의 성능을 보이기 위해 여러가지 실험을 수행한다. 이로부터 컨벌루션 신경망을 이용하는 우리의 검사 알고리즘이 기존의 기계학습에 쓰이는 분류 방법을 이용한 검사보다 결함이 있는 모터를 더 잘 구별해내는 것을 보인다. 특히 컨벌루션 신경망을 이용한 알고리즘은 종단간 학습의 일종으로 주어진 데이터에 특화된 특징을 사람이 직접 추출하지 않고도 뛰어난 성능을 보인다. 따라서 주어진 데이터에 대한 깊은 이해가 필요없이 음향 측정 데이터를 이용하는 다양한 검사 분야에 적용이 가능하다.

주요어: 컨벌루션 신경망, 전이 학습, 탐욕적 층별 지도 학습, 전기 모터 검사, 종단간 학습

학번: 2015-20734